11 PROGRAMMABLE TIMERS AND I/O PORTS

The processor has two identical timer blocks, each of which has two basic functions:

- Pulse Width Waveform Generation/ PWMOUT (PWMOUT mode)
- Pulse Width Count/Capture. (WIDTH_CNT mode)

You can configure the timer in either mode. The timer has one input/output pin—PWM_EVENTx. This pin functions as an output pin in the PWMOUT mode and as an input pin in the WIDTH_CNT mode. To implement these functions, each timer has three registers—TPERIODx, TPWIDTHx, and TCOUNTx.

All timer counters are 32-bits wide and use the processor's 2xCLKIN internal clock, which evaluates to a maximum period of 71.5 sec ($(2^{32}-1)$ * 16.67 ns internal clock cycles) for the timer count.

To enable or disable the timer, you set or clear the TIMENx bit in the MODE2 register. Figure 11-1 on page 11-2 shows the timer's enable and disable timing.

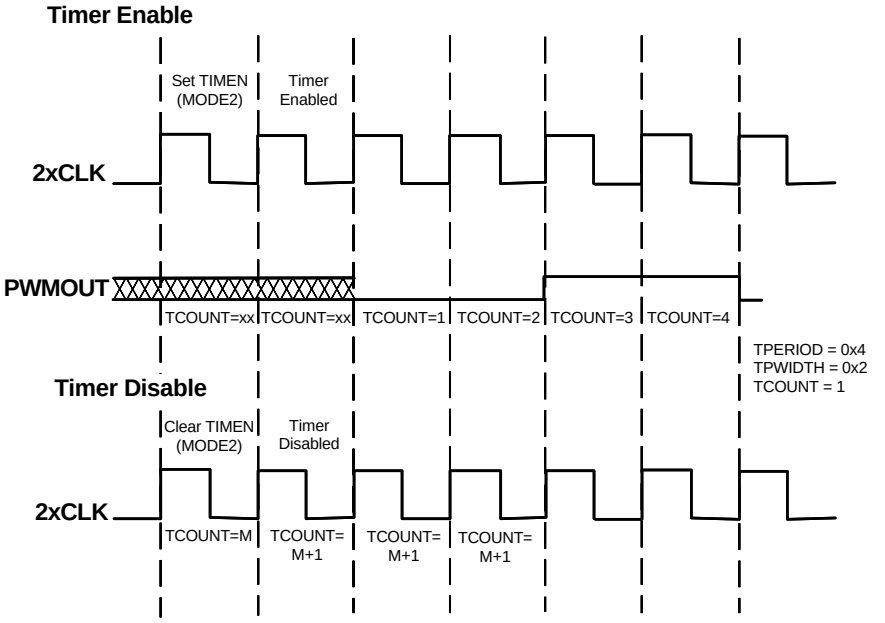

Figure 11-1. Timer enable and disable timing

PWMOUT Mode

In PWMOUT mode, the PWM_EVENTx is an output pin. To select it, you set the PWMOUTx bit high in the MODE2 register. The registers TPERIODx and TPWIDTHx contain the values of the timer count period and PWM output pulse width respectively.

To avoid unpredictable results of the PWM_EVENTx signal:

- Initialize TPWIDTHx and TPERIODx before enabling the timer.
- Do not alter TPWIDTHx and TPERIODx while the timer is enabled.
- Make sure the value of TPWIDTHx is less than the value of TPERIODx.

When the timer is enabled in this mode, the PWM_EVENTx is pulled low each time the TCOUNTx (up counter) value equals the TPERIODx value, and it is pulled high when the TCOUNTx value equals the TPWIDTHx value. TCOUNTx is reset once to `0x0000 0001` when the timer is enabled and each time TCOUNTx reaches the TPERIODx value. See Figure 11-1 on page 11-2.

When TCOUNTx equals TPERIODx, a timer interrupt (if enabled) is generated, and the CNT_EXPx/CNT_OVFx bit in the STKY register is set. The CNT_EXPx/CNT_OVFx bit is a sticky bit, and software must reset it explicitly. At reset, its value is `0`. Figure 11-2 shows the timer flow.

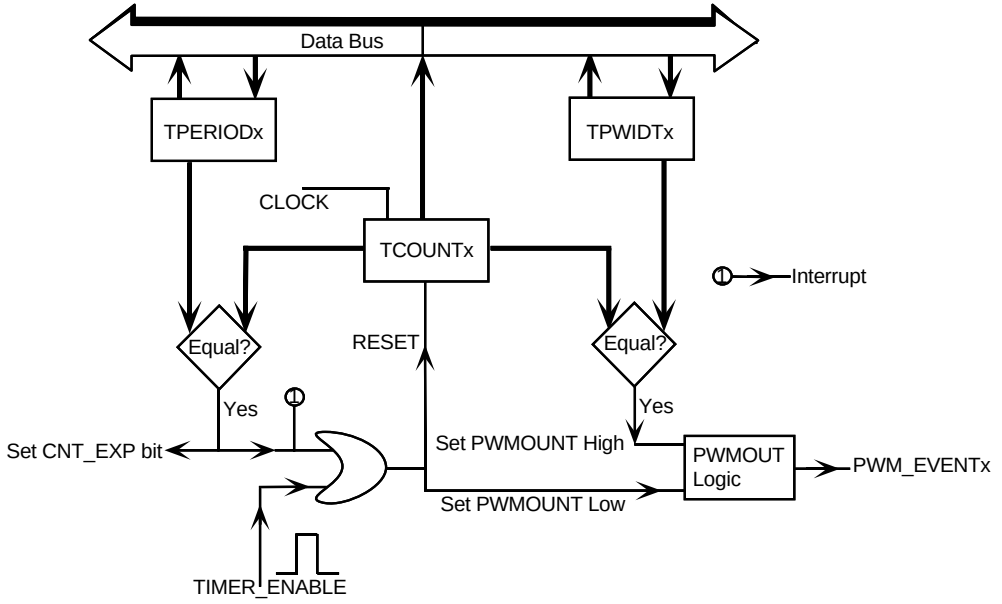

Figure 11-2. Timer Flow Diagram–PWMOUT Mode

WIDTH_CNT Mode

In the WIDTH_CNT mode, the PWM_EVENTx is an input pin. To select this mode, you set the PWMOUTx bit low in the MODE2 register.

When enabled in this mode, the timer resets TCOUNTx to `0x0000 0001` when it detects the leading edge of the PWM_EVENTx pin and starts counting (increments).

When it detects the trailing edge, the timer captures the current value of the TCOUNTx into the TPWIDTHx register. At the leading edge, the timer transfers the current value of the TCOUNTx into the TPERIODx register. This timing, shown in Figure 11-3, assumes the leading edge is set as `0` → `1`.

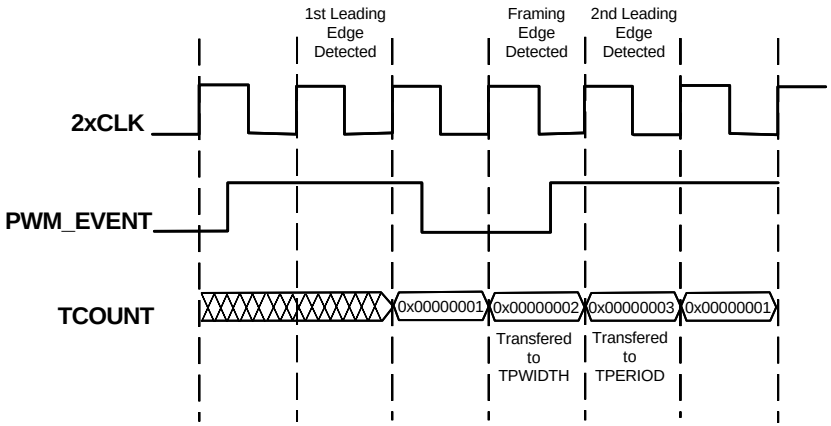

Figure 11-3. WIDTH_CNT mode timing

In this case, your software application can measure both the pulse width and the pulse period values, which are available in the TPWIDTHx and the TPERIODx registers, respectively.

To control the definition of leading edge and trailing edge of the PWM_EVENTx, you set the PULSE_HIx bit in the MODE2 register.

TPERIODx and TPWIDTHx are read-only registers when the timer is enabled in WIDTH_CNT mode.

A timer interrupt (if enabled) is generated when the timer captures either the pulse width or the pulse period value, which depends on the value of the PERIOD_CNTx bit in the MODE2 register.

If the PERIOD_CNTx is set high, the interrupt and the PULSE_CAPx bits (in the STKY register) get set when the pulse period value is captured. If the PERIOD_CNTx is set low, then the interrupt and the PULSE_CAPx are set when the pulse width value is captured.

A timer interrupt (if enabled) is also generated if the counter TCOUNTx reaches a value of `0xFFFF FFFF`:

- Before the edge for the pulse period is detected if PERIOD_CNTx is high
- Before the edge for the pulse width is detected if the PERIOD_CNTx is low.

In addition, the status bit CNT_EXPx/CNT_OVFx in the STKY register is set, indicating that TCOUNTx overflowed before the timer counted the maximum (2^{32}–2) intervening clock cycles.

PULSE_CAPx and CNT_EXPx/CNT_OVFx are sticky bits, and software has to explicitly clear them.

Note that the TPERIODx, TPWIDTHx and TCOUNTx (`x=0,1`) are all IOP memory mapped registers, not universal registers. Figure 11-4 on page 11-7 shows the timer flow.

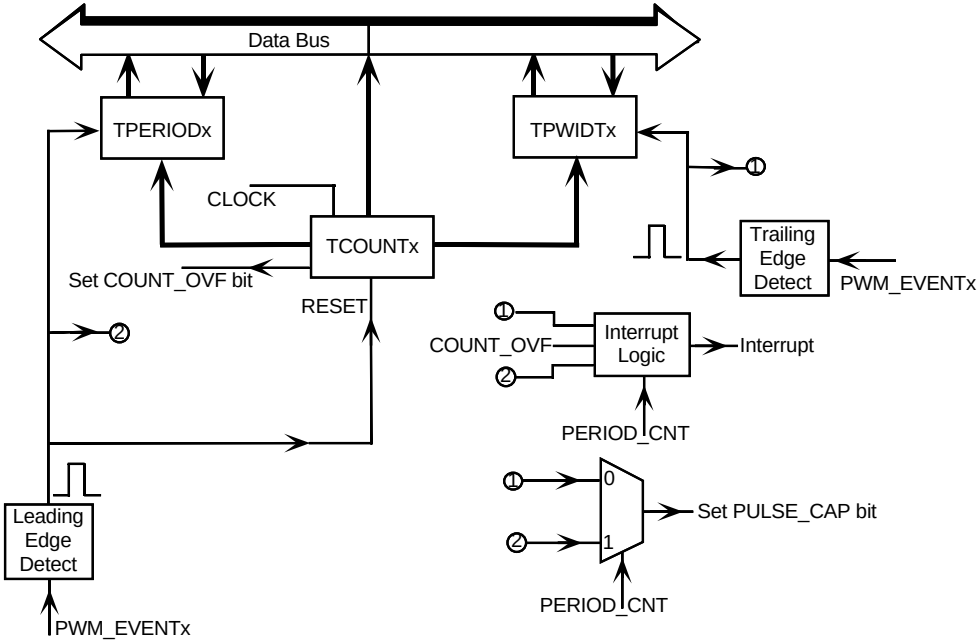

Figure 11-4. Timer Flow Diagram–WIDTH_CNT Mode

Timer Control Bits and the Interrupt Vectors

This section describes the timer control bit definitions and the MODE2 register definitions.

TIMENx
Timer enable (x=0,1)

0 = Disable

1 = Enable

PWMOUTx
PWMOUT/WIDTH_CNT control (x=0,1)

1 = PWM_EVENT is a PWMOUT output.

0 = PWM_EVENT is an WIDTH_CNT input. (default)

PULSE_HIx (x=0,1)
Applies to the WIDTH_CNT mode only

0 = 0 to 1 transition is leading edge in the WIDTH_CNT mode.

1 = 1 to 0 transition is leading edge in the WIDTH_CNT mode.

PERIOD_CNTx
Enable period count (applicable only to the WIDTH_CNT mode)

0 = Enable width count.

Interrupt and the PULSE_CAPx bits are set when pulse width is captured.

1 = Enable period count.

Interrupt and the PULSE_CAPx bits are set when pulse period is captured.

INT_HIx
Interrupt vector location. (x=0,1)

The two timers generate interrupts, and these can be latched either at bit 4 (TMZHI) or at bit 23 (TMZLI) of the IRPTL register, as shown in Table 11-1. In addition, these interrupts can be masked using the IMASK register.

Table 11-1. Timer status

INT_HI1	INT_HI0	Status
0	0	Both timers latch to TMZLI
0	1	timer1 => TMZLI, timer0 => TMZHI
1	0	timer1 => TMZHI, timer0 => TMZLI
1	1	Both timers latch to TMZHI

Timer Interrupts and the Status Stack

Only the timer interrupt on the TMZHI bit pushes the status stack, so, in the above combinations, 00 will not push the status stack, but both 01 and 10 will push the status stack, depending on which timer is programmed to cause the TMZHI interrupt. When using the 11 combination, interrupts generated by either timer push the status stack.

When using the 00 and 11 combinations, the processor latches a logical OR function of the two timer interrupts into the interrupt latch register. The software checks the CNT_EXPx and the EDGE_CAPx bits, determines the source of the interrupt, and takes appropriate action.

Figure 11-5 on page 11-10 shows the mapping of the MODE2 register.

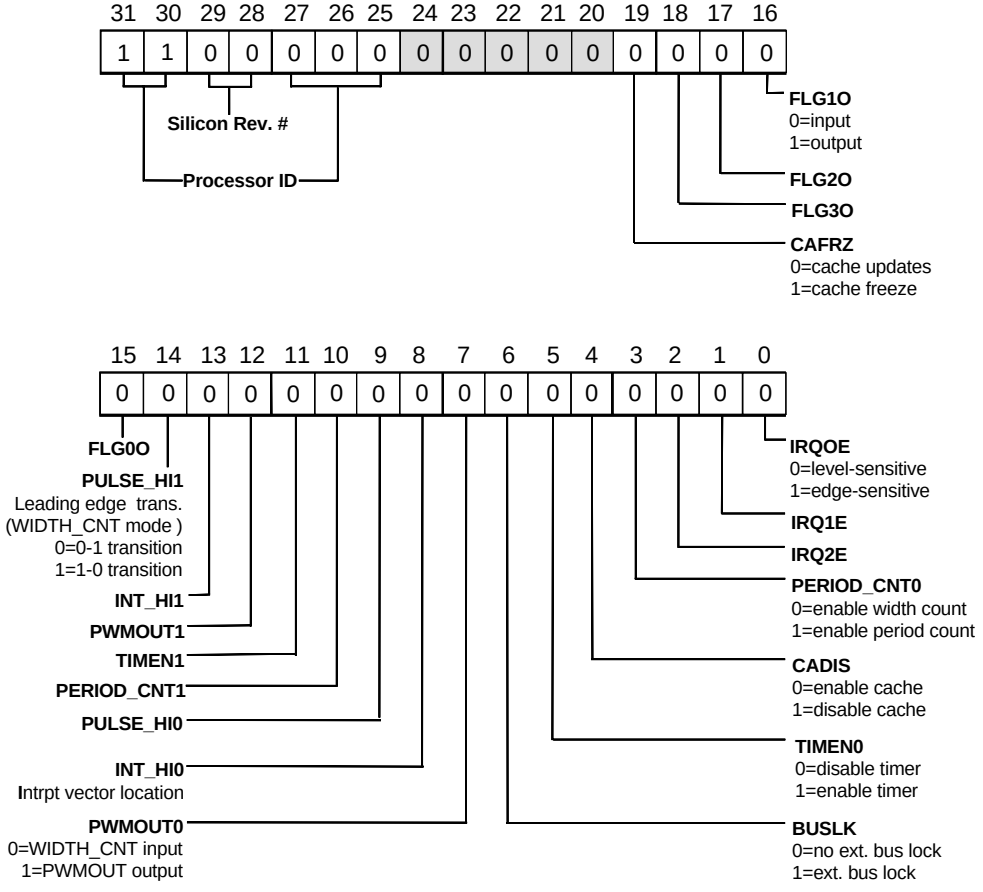

Figure 11-5. MODE2 Register

The STKY Register

Table 11-2 shows the CNT_EXPx and the PULSE_CAPx status bits in the STKY register.

Table 11-2. Timer status bits in the STKY register

Bit	Name	Description
12	PULSE_CAP0	Pulse captured bit for timer 0.
13	CNT_EXP0/CNT_OVF0	Counter expired/counter overflowed bit for timer 0.
14	PULSE_CAP1	Pulse captured bit for timer 1.
15	CNT_EXP1/CNT_OVF1	Counter expired/counter overflowed bit for timer 1.

Timer Registers and their Values at Reset

The TCOUNTx, TPWIDTHx, and TPERIODx registers are memory mapped. While TPERIODx and TPWIDTHx are read/write registers, TCOUNTx is read-only.

The timer enable signal gates the timer clock, interrupts, and the edge detect logic. In PWMOUT mode, TPWIDTHx and TPERIODx must be initialized before the timer is enabled. The timer is disabled at reset, and, at that time, TPERIODx, TCOUNTx and TPWIDTHx are unknown.

Table 11-3 summarizes the IOP register addresses for the timer registers:

Table 11-3. IOP register addresses

Register	Address
TPERIOD0	0x28
TPWIDTH0	0x29
TCOUNT0	0x2a
TPERIOD1	0x2b
TPWIDTH1	0x2c
TCOUNT1	0x2d

Programmable I/O Ports

The processor has twelve flag pins $FLAG_{11-0}$, which are programmable, general-purpose I/O ports.

The MODE2 register configures the functionality, or direction, of the pins $FLAG_{3-0}$, and the ASTAT register reflects the value of these flag bits.

The functionality of the $FLAG_{11-4}$ pins is similar to that of the $FLAG_{3-0}$, but the IOP registers IOCTL and IOSTAT contain their control and status bits.

You cannot execute the bitwise operations, such as BIT TST, BIT CLR, and so on, directly on the IOP registers. To perform these operations on the $FLAG_{4-11}$ pins, you must first transfer the contents of the IOSTAT register (shown in Figure 11-6) to the Register File or to another universal register.

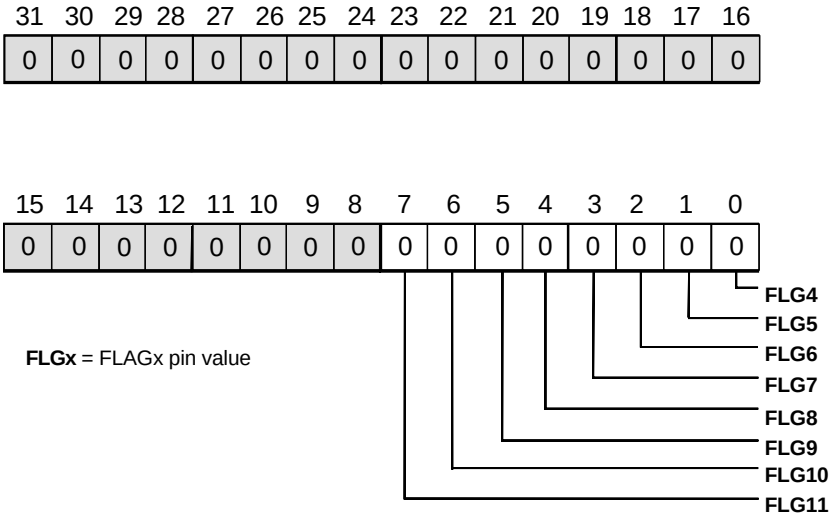

Figure 11-6. IOSTAT register

ASTAT is a universal register, so the status on $FLAG_{3-0}$ can be checked and manipulated using the bitwise operations directly. This is the difference between the $FLAG_{3-0}$ and the rest of the FLAG pins.

For detailed description of the IOCTL register, where the directions on the I/O ports are set, see Chapter 10, SDRAM Interface.

For a description of the IOSTAT register, see Figure 11-6.

The IOP address locations for the IOCTL and the IOSTAT registers are `0x2e` and `0x2f` respectively.